Texas A&M University - Commerce

Department of Computer Science

Comparison of SVM and Random Forests for Heart Disease Risk Prediction

Lakshmi Chandana Narra

Supervisor: Derek Harter, Ph.D.

A report submitted in partial fulfilment of the requirements of
Texas A&M University - Commerce for the degree of
Master of Science in *Computer Science*

February 24, 2024

Declaration

I, Lakshmi Chandana Narra, of the Department of Computer Science, Texas A&M University - Commerce, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of TAMUC and public with interest in teaching, learning and research.

Lakshmi Chandana Narra
February 24, 2024

Abstract

In recent times, the global rise in cardiovascular diseases has become increasingly prevalent, influenced by evolving lifestyles and societal factors. Emphasizing the need for timely detection and ongoing monitoring, particularly in regions with limited medical resources. Utilizing a public health dataset on patient heart health, including information from medical procedures and ongoing patient monitoring, this research uniquely centers on the comparative analysis of SVM and Random Forests. Focused on these two algorithms, this research aligns with the evolving landscape of machine learning in healthcare, presenting a concentrated perspective on their potential contributions. The methodology involves training SVM and Random Forest models on the dataset, evaluating their performance using key accuracy metrics such as the confusion matrix, Accuracy, precision, and F1 score. The study anticipates achieving comparable accuracy between the models but aims to determine their relative strengths in precision, recall, and F1 scores. This research aims to provide insights into which algorithm may be better suited for addressing the challenges associated with cardiovascular health monitoring, taking into consideration all parameters assessed in the research conclusion.

Keywords: Machine Learning, Random Forest, Support Vector Machine, Cardiovascular disease , Healthcare

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

List of Figures

List of Tables

List of Abbreviations

SVM	Support Vector Machine
RF	Random Forest

Chapter 1

Introduction

Cardiovascular diseases represent a formidable global health challenge, with their prevalence escalating and ranking among the primary causes of morbidity and mortality. The pressing concern is the need for robust predictive models to address the increasing burden of heart diseases, enabling early detection and effective risk mitigation strategies.

1.1 Background

This research project delves into the development and evaluation of predictive models for heart disease using machine learning algorithms, specifically focusing on Support Vector Machines (SVM) and Random Forests(RF). The scope encompasses a comprehensive analysis of these algorithms, exploring their capabilities and limitations in accurately predicting the risk of cardiovascular events. The context of the project revolves around leveraging a diverse dataset derived from various medical procedures and continuous patient monitoring to enhance our understanding of heart disease prediction.

The background of this study lies in the evolving landscape of lifestyle, dietary habits, and healthcare dynamics that contribute to the increasing prevalence of cardiovascular diseases. The significance of early detection and continuous monitoring underscores the importance of advanced Toma and Wei (2023)predictive modeling techniques. Against this backdrop, the research aims to contribute to the field of cardiovascular health by providing insights into the efficacy of SVM and Random Forest algorithms.

In Kumari et al. (n.d.) a comparative study on classification methods namely Ripper, Decision Tree, Artificial neural networks and Support Vector Machine are analyzed on cardiovascular disease dataset.

In Yanwei et al. (2007)it is establishes that a number of factors have been shown to increase the risk of developing heart disease. Some of these family history, high levels of LDL bad cholesterol, Family history of cardiovascular disease, High levels of LDL (bad) cholesterol, Low level of HDL (good) cholesterol, Hypertension, High fat diet, Lack of regular exercise, Obesity.

In summary, the investigated problem centers on the escalating prevalence of cardiovascular diseases, and the project's scope involves the development and evaluation of predictive models using SVM and Random Forest. The background highlights the contextual relevance of advanced

predictive modeling in addressing the challenges posed by heart diseases in the contemporary healthcare landscape.

1.2 Problem statement

The research question guiding this study is: "How do Support Vector Machines (SVM) and Random Forest algorithms differ in terms of accuracy, efficiency, and interpretability when predicting the risk of heart disease?"

The prevalence of cardiovascular diseases is increasing globally, necessitating accurate and efficient predictive models for early detection and intervention. However, selecting the most suitable algorithm for this task poses a challenge. This research aims to compare and examine the differential performance of SVM and Random Forests in predicting the risk of heart disease. By leveraging real-world dataDavid Lapp (1988) on patient heart health, the study seeks to uncover the unique strengths and limitations of each algorithm. Through this investigation, the research aims to provide insights into selecting appropriate machine learning algorithms to enhance cardiovascular health monitoring and decision-making in clinical practice.

1.3 Aims and objectives

This research project's main goal is to evaluate and contrast the effectiveness of Random Forest and Support Vector Machines (SVM) algorithms in relation to risk assessments for heart disease. The main objective is to advance predictive modeling methods for accurate assessment and early identification of cardiac disease.

- Analyze and clean the Kaggle heart disease dataset, preparing it for building predictive models.
- Develop a Support Vector Machine (SVM) classifier and train it on the preprocessed heart disease data.
- Build a Random Forest (RF) classifier and train it on the same preprocessed data.
- Conduct parameter tuning to maximize the performance of both models, focusing on recall, precision, and/or F1 scores.
- Compare the recall, precision, and accuracy of the resulting SVM and RF models.
- Evaluate and interpret the performance of SVM and RF models in predicting heart disease risk.
- Identify the strengths and weaknesses of each model in the context of cardiovascular health monitoring.
- Draw conclusions and provide recommendations for selecting the most suitable machine learning algorithm for heart disease risk analysis.

1.4 Solution approach

Data Preparation: Prepare the Kaggle heart disease dataset for the construction of predictive models by analyzing and cleaning it.
Model Development: Using the preprocessed cardiac disease data, create an SVM classifier and train it. Using the same preprocessed data, create and train a Random Forest (RF) classifier.
Model Assessment: Optimize both models' performance by fine-tuning their parameters with an emphasis on recall, precision, and/or F1 scores.

Print the assessment measures for each of the two models:

- Metrics for the SVM Model: F1 Score, Accuracy, Precision, Recall.
- Metrics for the Random Forest Model: F1 Score, Accuracy, Precision, Recall.

Print the two models' confusion matrices.
Model Comparison: Based on the evaluation metrics and confusion matrices, compare the SVM and Random Forest models' performances. Analyze and assess how well the RF and SVM models predict the risk of heart disease. Determine each model's advantages and disadvantages in relation to cardiovascular health monitoring.
Concluding remarks and suggestions:

- Make inferences from the performance comparison.
- Make suggestions on which machine learning algorithm would be best for analyzing the risk of heart disease.

The measures used to answer the research topic of comparing and contrasting the efficiency of Random Forest and SVM algorithms for heart disease risk assessments are described in this solution methodology. Data preparation, model construction, assessment, comparison, and conclusion are all included, giving your research a thorough approach.

1.5 Summary of contributions and achievements

Chapter 2

Literature Review

This section explores the efficacy of machine learning (ML) in the prediction of cardiovascular disease (CVD). ML learns from data and experience through training, enabling it to be applied to various tasks based on specific algorithms. This flexibility enables ML algorithms to analyze complex datasets and predict CVD risk.

A review of existing literature is also done to investigate previously published studies in the area. This review helps to contextualize the current findings. The literature review for this study will look at earlier research on machine learning (ML) in disease prediction, including various algorithms and their effectiveness. This understanding is crucial for developing accurate and effective predictive models for CVD.

2.1 Example of in-text citation of references in LaTeX

The references in a report relate your content with the relevant sources, papers, and the works of others. To include references in a report, we *cite* them in the texts. In MS-Word, EndNote, or MS-Word references, or plain text as a list can be used. Similarly, in LaTeX, you can use the "thebibliography" environment, which is similar to the plain text as a list arrangement like the MS word. However, In LaTeX, the most convenient way is to use the BibTex, which takes the references in a particular format [see references.bib file of this template] and lists them in style [APA, Harvard, etc.] as we want with the help of proper packages.

These are the examples of how to *cite* external sources, seminal works, and research papers. In LaTeX, if you use "**BibTex**" you do not have to worry much since the proper use of a bibliographystyle package like "agsm for the Harvard style" and little rectification of the content in a BiBText source file [In this template, BibTex are stored in the "references.bib" file], we can conveniently generate a reference style.

Take a note of the commands \cite{} and \citep{}. The command \cite{} will write like "Author et al. (2019)" style for Harvard, APA and Chicago style. The command \citep{} will write like "(Author et al., 2019)." Depending on how you construct a sentence, you need to use them smartly. Check the examples of **in-text citation** of sources listed here [This template recommends the **Harvard style** of referencing.]:

- **?** has written a comprehensive guide on writing in LaTeX [Example of \cite{}].

- If LaTeX is used efficiently and effectively, it helps in writing a very high-quality project report (**?**) [Example of \citep{}].
- A detailed APA, Harvard, and Chicago referencing style guide are available in (**?**).

Example of a numbered list:

1. **?** has written a comprehensive guide on writing in LaTeX.
2. If LaTeXis used efficiently and effectively, it helps in writing a very high-quality project report (**?**).

2.2 Example of "risk" of unintentional plagiarism

Using other sources, ideas, and material always bring with it a risk of unintentional plagiarism. **MUST**: do read the university guidelines on the definition of plagiarism as well as the guidelines on how to avoid plagiarism (Kumar et al., 2018).

2.3 Critique of the review

Describe your main findings and evaluation of the literature.

2.4 Summary

Write a summary of this chapter

Chapter 3

Methodology

We mentioned in Chapter 1 that a project report's structure could follow a particular paradigm. Hence, the organization of a report (effectively the Table of Content of a report) can vary depending on the type of project you are doing. Check which of the given examples suit your project. Alternatively, follow your supervisor's advice.

3.1 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project's needs.

3.1.1 Example of a software/Web development main text structure

Notice that the "methodology" Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that "Chapter 5" in Table 3.2 is "Testing and Validation" which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

3.1.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

3.1.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Table 3.1: Undergraduate report template structure

Frontmatter		Title Page Abstract Acknowledgements Table of Contents List of Figures List of Tables List of Abbreviations
Main text	Chapter 1	Introduction
	Chapter 2	Literature Review
	Chapter 3	Methodology
	Chapter 4	Results
	Chapter 5	Discussion and Analysis
	Chapter 6	Conclusions and Future Work
	Chapter 7	Refection
End matter		References Appendices (Optional) Index (Optional)

Table 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Requirements specifications
		Analysis
		Design
		Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

3.1.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the "Methodology" section as "Materials and Methods."

Table 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Algorithms descriptions
		Implementations
		Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Problems (tasks) descriptions
		Algorithms/tools/technologies/etc. descriptions
		Implementations
		Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

Table 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	
		Problems (tasks) description
		Materials
		Procedures
		Implementations
		Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.2 Example of an Equation in LaTeX

Eq. 3.1 [note that this is an example of an equation's in-text citation] is an example of an equation in LaTeX. In Eq. (3.1), s is the mean of elements $x_i \in \mathbf{x}$:

$$s = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{3.1}$$

Have you noticed that all the variables of the equation are defined using the **in-text** maths command $.$, and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

3.3 Example of a Figure in LaTeX

Figure 3.1 is an example of a figure in LaTeX. For more details, check the link:
wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.
Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the "Flow algorithm software". Here is the link for that: flowgorithm.org.

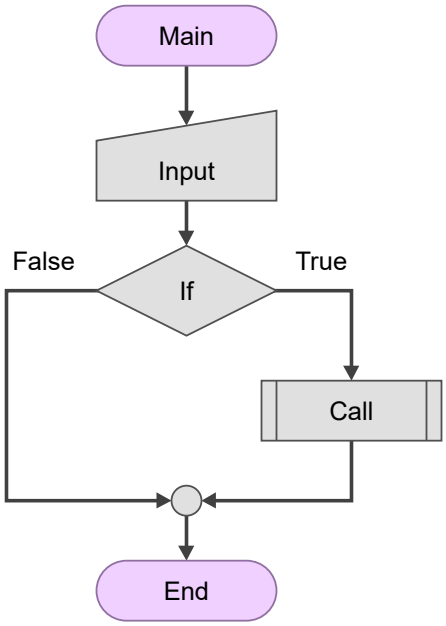

Figure 3.1: Example figure in LaTeX.

3.4 Example of an algorithm in LaTeX

Algorithm 1 is a good example of an algorithm in LaTeX.

Algorithm 1 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \ldots, x_N$
Output: $EvenSum$ (Sum of even numbers in $\mathbf{x}$)

1: **function** EvenSummation($\mathbf{x}$)
2: $EvenSum \leftarrow 0$
3: $N \leftarrow length(\mathbf{x})$
4: **for** $i \leftarrow 1$ to N **do**
5: **if** $x_i \mod 2 == 0$ **then** ▷ check if a number is even?
6: $EvenSum \leftarrow EvenSum + x_i$
7: **end if**
8: **end for**
9: **return** $EvenSum$
10: **end function**

3.5 Example of code snippet in LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/-scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```python
import numpy as np

x  = [0, 1, 2, 3, 4, 5] # assign values to an array
evenSum = evenSummation(x) # call a function

def evenSummation(x):
    evenSum = 0
    n = len(x)
    for i in range(n):
        if np.mod(x[i],2) == 0: # check if a number is even?
            evenSum = evenSum + x[i]
    return evenSum
```

Listing 3.1: Code snippet in LaTeX and this is a Python code example

Here we used the "\clearpage" command and forced-out the second listing example onto the next page.

```java
public class EvenSum{
    public static int evenSummation(int[] x){
        int evenSum = 0;
        int n = x.length;
        for(int i = 0; i < n; i++){
            if(x[i]%2 == 0){ // check if a number is even?
                evenSum = evenSum + x[i];
            }
        }
        return evenSum;
    }
    public static void main(String[] args){
        int[] x  = {0, 1, 2, 3, 4, 5}; // assign values to an array
        int evenSum = evenSummation(x);
        System.out.println(evenSum);
    }
}
```

Listing 3.2: Code snippet in LaTeX and this is a Java code example

```cpp
int evenSummation(int x[]){
    int evenSum = 0;
    int n = sizeof(x);
    for(int i = 0; i < n; i++){
        if(x[i]%2 == 0){ // check if a number is even?
            evenSum = evenSum + x[i];
      }
    }
    return evenSum;
}

int main(){
    int x[]  = {0, 1, 2, 3, 4, 5}; // assign values to an array
    int evenSum = evenSummation(x);
    cout<<evenSum;
    return 0;
}
```

Listing 3.3: Code snippet in LaTeX and this is a C/C++ code example

3.6 Example of in-text citation style

3.6.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more them once.

3.6.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase "Eq" for an equation before using an equation number with (\eqref{.}). Use "Table" to refer to a table, "Figure" to refer to a figure, "Algorithm" to

refer to an algorithm and "Listing" to refer to listings (code snippets). Note that, we do not use the articles "a," "an," and "the" before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence "A report structure is shown in **the** Table 3.1" should be written as "A report structure is shown **in** Table 3.1."

3.7 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter "**Testing and Validation**" should precede the "Results" chapter. See Section 3.1.1 for report organization of such project.

Chapter 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in LaTeX

Table 4.1 is an example of a table created using the package LaTeX"booktabs." do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of liens (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Table 4.1: Example of a table in LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork) goes below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table goes above** the table. See the example in Table 4.1.
- The **caption of an Algorithm goes above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing goes below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with "Results" Chapter as " Results and Discussion" as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

Discussion and analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader's understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summaries the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most signification contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. The future work is then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Chapter 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in LaTeXand use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planed aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is "**References**," which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between "References" and "Bibliography." The list of "References" strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of "References."

References

David Lapp (1988), 'Public health dataset for heart disease prediction'.
URL: *https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset*

Kumar, P. P., Chandana, N. L., Reddy, A. S. V. M. and Rao, D. R. (2018), 'A comprehensive analysis on sentimental data set using machine learning technique', *International Journal of Mechanical Engineering and Technology (IJMET)* **9**(1), 320–326.
URL: *http://iaeme.com/Home/issue/IJMET?Volume=9Issue=1*

Kumari, M., Godara, S. and Kinoshita, A. (n.d.), 'Comparative study of data mining classification methods in cardiovascular disease prediction', *IJCST* **2**, 304–308.

Toma, M. and Wei, O. C. (2023), 'Predictive modeling in medicine', *Encyclopedia* **3**(2), 590–601.

Yanwei, X., Wang, J., Zhao, Z. and Gao, Y. (2007), Combination data mining models with new medical data to predict outcome of coronary heart disease, *in* 'Proceedings International Conference on Convergence Information Technology', pp. 868–872.

Appendix A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Appendix B

An Appendix Chapter (Optional)

...